

GUIDANCE ENHANCEMENTS AND PERFORMANCE ASSESSMENTS FOR THE MARS ASCENT VEHICLE SPIN-STABILIZED UPPER STAGE CONFIGURATION

Jason Everett*, Dane Erickson†, Joseph Powers‡

The objective of the Mars Sample Return (MSR) campaign is to return samples from the surface of Mars to Earth for research. As one element of the MSR campaign, the Mars Ascent Vehicle (MAV) is responsible for transporting the samples from the surface of Mars to a Low-Martian Orbit (LMO) for retrieval. Complete autonomy is required throughout ascent, and orbital insertion is constrained by tight dispersion boundaries. An unguided, spin-stabilized second stage for MAV has been selected over a guided upper-stage to drive mass savings and reduce overall MSR campaign mass risk, at the cost of reduced GNC capability. To address this design change, the MAV GNC team has derived a robust prediction algorithm, building on previous energy management schemes, that solves for a single inertial pointing direction solution for the spin-stabilized 2nd stage burn. Algorithm stability is explored that compared to previous versions of the algorithm. Also, a set of analytical partials was developed to study MAV's dispersed orbital insertion performance with respect to MAV system uncertainties. These partials were verified through simulation analysis and prove useful for analytical insight into the dynamics of MAV during the 2nd stage maneuver.

INTRODUCTION

The objective of the Mars Sample Return (MSR) campaign is to collect samples from the surface of Mars and return the samples to Earth for research and analysis. As one element of the MSR campaign, the Mars Ascent Vehicle (MAV) is responsible for transporting the Mars samples from the surface of Mars to a Low-Martian Orbit (LMO) for retrieval. The current MAV architecture is a two-stage solid motor vehicle in which the first stage contains active control, avionics and GNC, and the second stage is unguided and spin-stabilized. The MAV vehicle will be transferred to the Martian surface by the Mars Lander Platform (MLP), currently being designed by NASA's Jet Propulsion Laboratory (JPL). The MAV will be stowed aboard the MLP horizontally, both for easy access into the payload bay for sample collection and for greater stability throughout Martian/Earth aerodynamic regimes and launch environments. Once on the Martian surface, samples will be transferred from either a Sample Fetch Rover (SFR) or the Perseverance Rover (currently on the Martian surface) into the MAV upper stage via a Sample Transfer Arm (STA). Samples will be contained in up to 30 Returnable Sample Tube Assemblies (RSTAs) allowing for up to 0.5kg of returned Martian regolith^{1,2}.

* Aerospace Engineer, NASA Marshall Space Flight Center, EV42/Guidance, Navigation & Mission Analysis

† Aerospace Engineer, NASA Marshall Space Flight Center, EV42/Guidance, Navigation & Mission Analysis

‡ Aerospace Engineer, NASA Marshall Space Flight Center, EV41/Control Systems Design & Analysis

Once the samples are loaded onto the MAV, it is the responsibility of the MAV to launch and insert the Orbiting Sample (OS) payload into a stable and predictable LMO. Immediately before MAV's 1st stage ignition, the MLP will "throw" the MAV into the air by the JPL-designed Vertical Egress Controlled Tip Off Rate (VECTOR) launch system. After this initial launch from the lander, the MAV has complete autonomy of its 1st stage ignition and flight, coast, fairing separation, and orbital insertion burn. Once in LMO, a vehicle designed by the European Space Agency (ESA) named the Earth Return Orbiter (ERO) will transfer from Earth to a MAV-matching LMO, collect the OS, and return to Earth. The upper stage of the MAV vehicle will remain in LMO and decay over time. Extensive literature can be found on the design iterations of MSFC's MAV design ^{1,2,3}. Figure 1 shows the nominal ConOps of the MAV ascent flight profile.

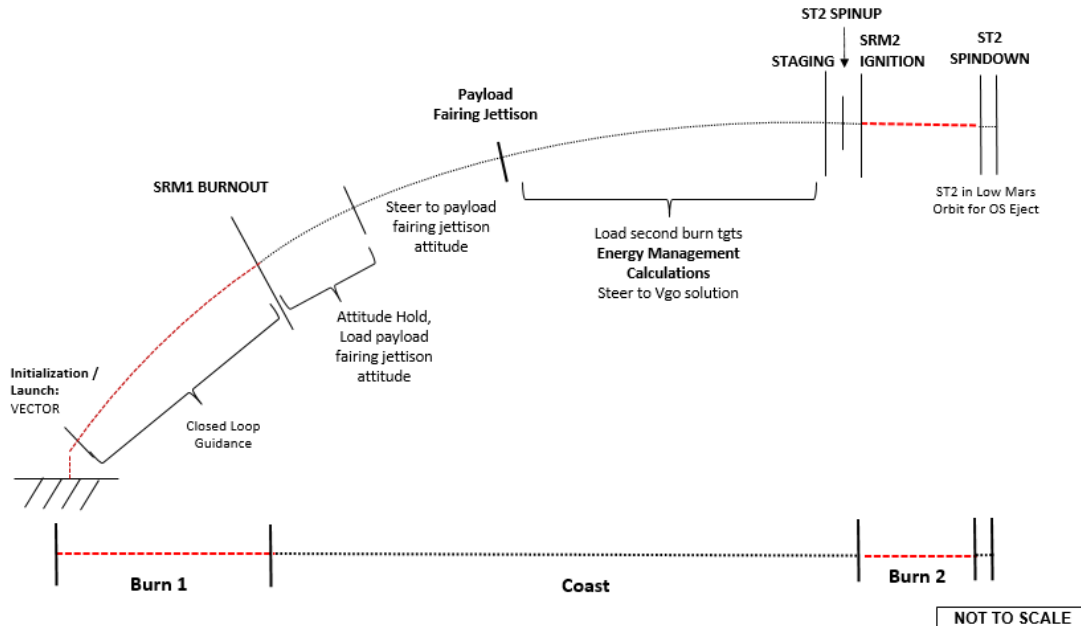


Figure 1: MAV ConOps Ascent Flight Profile

The MAV GNC Team

NASA's Marshall Space Flight Center (MSFC) is responsible for the design definition and software development of the MAV. Within the MSFC MAV team, the MAV Guidance, Navigation and Controls (GNC) team consists of a group of engineers that are responsible for designing algorithms and control laws that will allow the MAV to achieve successful orbital insertion within expected dispersed bounds given a range of variable conditions experienced during Mars ascent. This includes navigation filters, navigation propagation routines for vehicle state prediction, guidance orbital targeting routines, guidance command steering filters, PID control laws for thrust vector control (TVC) systems, phase plane control laws for reaction control systems (RCS), and several other algorithms. The MAV GNC team is also responsible for simulating the flight dynamics of the vehicle to test the efficiency of these GNC algorithms in multiple 6-degree-of-freedom (6DOF) simulation suites. The MAV GNC team is responsible for incorporating software models of all vehicle components from other design groups at MSFC into these 6DOF simulations. The MAV GNC team then acts as a verification group for verifying several requirements by analysis through these simulations. Once algorithms are fully defined and tested for the MAV ascent flight profile, the MAV GNC team interfaces with the MAV flight software (FSW) team at MSFC to ensure that the algorithms are coded properly and efficiently on-board MAV's 1st stage flight processor. The MAV FSW team also maintains a small standalone physics simulation that can be used for verification against the MAV GNC team's 6DOF simulations.

The Unguided Upper-Stage MAV Configuration

Preliminary iterations of MSFC's MAV design included active TVC and RCS control on both first stage and second stage ^{1,2,3}. However, to meet updated mass constraints from the lander unit, a significant reduction in Gross-Liftoff-Mass (GLOM) drove a design change to an unguided, spin-stabilized second stage. This redesign allowed for significant mass reductions across the entire vehicle stack (no need for a TVC system or associated power components on the 2nd stage). To support this redesign, the RCS and avionics systems were moved to the 1st stage to support a lighter second stage after 1st stage separation. Instead of a TVC-controlled 2nd stage, the new MAV design now contains small spin-up and spin-down solid-propellant motors to control a high-rate spin of the 2nd stage, to spin-stabilize the orbital insertion burn. In lieu of avionics, the MAV 2nd stage was redesigned to be equipped with an embedded time-based sequencer that allows for simple timing of events (spin motor ignitions, 2nd stage motor ignition, etc.), but does not allow for any navigational awareness or more complex software prediction or control algorithms.

This modification posed many challenges to the MAV GNC team, with a significant challenge being the required modifications for the 2nd stage maneuver orbital targeting. Historically, the MAV GNC team has maintained an in-house targeting algorithm for MAV that has been referred to commonly as MAV's Energy Management algorithm ⁴. Results of integrated analysis utilizing an earlier version of this algorithm on a guided 2nd stage MAV can be found in a previous publication written by the MAV GNC team ⁷. With an unguided, spin-stabilized upper stage, the MAV vehicle can no longer receive closed-loop data on target updates *during* the 2nd stage burn, and rather must select an inertially-fixed orientation for the entirety of the 2nd stage maneuver. This immediately posed challenges to the current derivation of the algorithm and posed risk to the orbital insertion performance due to increased dispersions. The MAV GNC team was now tasked with modifying the insertion targeting algorithm to ensure that orbital insertion can be met with an inertially fixed orientation. The derivation of this new, enhanced algorithm is outlined in the following sections of this paper.

Analytical Insight on Orbital Dispersions

The shift to a spin-stabilized upper stage design for the MAV came with many increased dispersions due to the inability to control pointing orientation during the 2nd stage maneuver. When implementing early iterations of the unguided 2nd stage configuration, the MAV GNC team had noticed specific dispersion parameters leading to higher sensitivities in orbital insertion dispersions than other parameters. There was a clear distinction between 1st-order and 2nd-order effects on the final orbital insertion performance given specific dispersion parameters. The MAV GNC team had therefore decided to conduct an analytical study to characterize these effects through mathematics rather than pure simulation.

The results of this study were a set of analytical 1st-order partial derivatives of MAV's orbital insertion performance with respect to high-sensitivity vehicle dispersion inputs. These partials provided the team valuable insight into predicting how our dispersions will grow and change with respect to future vehicle design iterations and future dispersion changes. These partials were also compared to 6DOF simulation results, acting as a cross-verification of both the simulation and the partial derivations.

Paper Overview

This paper describes enhancements and modifications that were made to the MAV GNC algorithms to ensure that the MAV vehicle remains capable of successful orbital insertion with an unguided, spin-stabilized 2nd stage. A derivation of the new Energy Management prediction algorithm for 2nd stage targeting is firstly presented, which describes how to enable a fixed-orientation MAV 2nd stage orbital insertion burn. Notes on algorithm stability and mechanization are also presented that provide the reader with a comparison to the previous version of the targeting algorithm. Next, a set of analytical partials is presented that gives insight into the relative sensitivities of an unguided, inertially fixed 2nd stage maneuver. Lastly, these partials are compared to 6DOF simulated analysis for verification.

ENHANCED ENERGY MANAGEMENT ALGORITHM

At the heart of MAV's 2nd stage orbital insertion is an "Energy Management" targeting algorithm that has been derived in-house at MSFC. This algorithm allows the MAV 2nd stage SRM to target a specific set of orbital parameters at insertion given only a fixed amount of Δv on-board the 2nd stage. The algorithm itself has been through several iterations, and a more thorough background of the origins and history of the algorithm can be found in previous publications⁴. This section presents an overview of the fundamentals of the Energy Management algorithm, and the enhancements necessary to use this algorithm as an inertially-fixed targeting solution for a spin-stabilized upper stage.

The concept of "energy management" has been described in several different literature sources as an algorithm used to hit a specified orbital or ground target with a fixed on-board amount of energy^{8,9,10}. However, there has not yet been a two-stage SRM planetary ascent vehicle mission that has flown before that is tasked with successful orbital insertion from launch with similar energy constraints. The MAV GNC team has worked with several subject matter experts in the field to derive an algorithm that can achieve the desired orbital accuracy for the MAV program given specified energy constraints.

MAV's Energy Management algorithm works via a concept named *r-dot nulling*, which is represented visually in Figure 2. In MAV's *r-dot nulling* scheme, the final orbital state of the MAV after the 2nd stage burn is constrained to lie on the apsis line of the final orbit, thus eliminating any radial velocity at orbital insertion. This elimination of radial velocity can be thought of as a purposeful "energy bleed". The energy required for this radial energy bleed-off will decrease over the course of a ballistic coast up to apoapsis and will be zero at apoapsis. The MAV vehicle then must compute the time at which the amount of energy required to achieve orbital insertion equates to the estimated amount of energy on-board the 2nd stage.

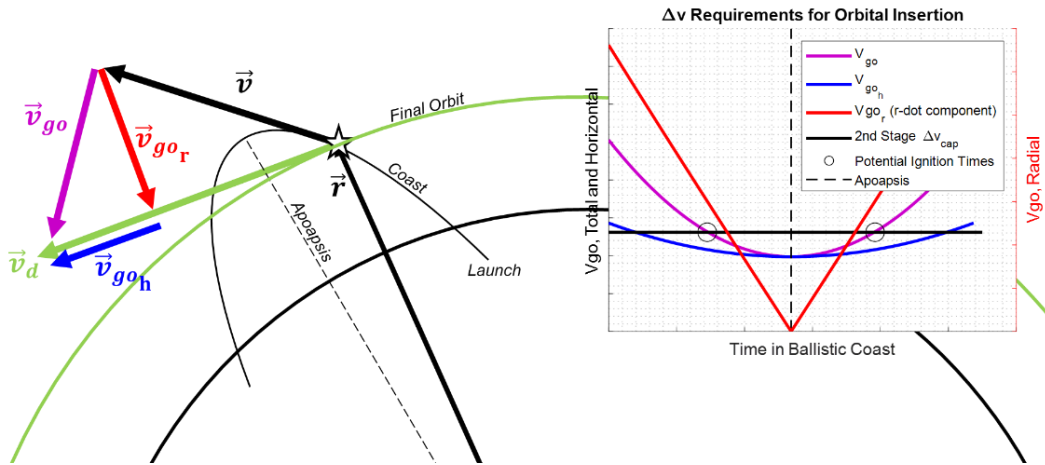


Figure 2 – The \dot{r} -Nulling Effect of Energy Management

This computation of time at which "energy matching" occurs is not a closed-form solution, and therefore requires iteration. The Energy Management algorithm can be thought of as two components: the inner loop (the *targeting algorithm*), and the outer loop (the *prediction algorithm*). The inner loop is a targeting algorithm which is responsible for computing how much Δv is required to perform orbital insertion at a specific state. The outer loop is an iterative scheme that finds the time(s) during ballistic coast at which the inner loop solution will return a Δv that is equal to the amount of Δv on-board the second stage. The inner-loop targeting algorithm and the outer-loop iteration scheme have both been modified since previous versions to address unguided/inertially fixed orientation requirements as well as stability/convergence concerns, and these modifications are addressed in the following sections.

The Inner Loop: Simple Cross-Product Steering (SxS)

A quick outline of the targeting routine used within the Energy Management algorithm framework is provided below, with a more extensive overview provided in the previous publication on SxS⁴. The SxS targeting routine, put plainly, tries to answer the following question: *given an initial state $\vec{X}_0 = [\vec{r}_0, \vec{v}_0]$, what is the final desired velocity required \vec{v}_d (and subsequently a total impulsive velocity-to-be-gained $\vec{v}_{go} = \vec{v}_d - \vec{v}_0$) such that the final state $\vec{X}_f = [\vec{r}_f, \vec{v}_d]$ meets a specified set of orbital requirements?*

The MAV final orbital insertion is constrained primarily by final semi-major axis. The secondary constraints are orbital inclination and right ascension of the ascending node (RAAN), to control final insertion orbital plane and plane drift. This brings the MAV to three parameters that are controlled as targets for the SxS targeting routine and are referenced through this paper as desired semi-major axis a_d , desired inclination i_d , and desired RAAN Ω_d . Desired final argument of periapsis ω_d and true anomaly f_d are constrained inherently through the concept of r-dot nulling. This leads to one free parameter: final orbital eccentricity. Indeed, the freedom of this final orbital parameter is what allows MAV to achieve a final desired semi-major axis in the presence of a fixed Δv_{cap} . Simulation results show that the resulting range of final orbital eccentricities is within acceptable bounds for the MSR campaign.

To answer the question of solving for \vec{v}_{go} , a special guidance “target frame” is defined that enables proper orientation of the solution \vec{v}_{go} vector. This frame is the $\hat{i}_x \hat{i}_y \hat{i}_z$ frame and is derived in Equations (1) through (3) as follows:

$$\hat{i}_y = \begin{bmatrix} -\sin(\Omega_d) \sin(i_d) \\ \cos(\Omega_d) \sin(i_d) \\ -\cos(i_d) \end{bmatrix} \quad (1)$$

$$\hat{i}_x = \frac{\vec{r} - (\vec{r} \cdot \hat{i}_y) \hat{i}_y}{|\vec{r} - (\vec{r} \cdot \hat{i}_y) \hat{i}_y|} \quad (2)$$

$$\hat{i}_z = \frac{\hat{i}_x \times \hat{i}_y}{|\hat{i}_x \times \hat{i}_y|} \quad (3)$$

As seen in Equations (1) through (3), \hat{i}_x represents the current position vector projected onto the target orbital plane, \hat{i}_y represents the orbit plane itself (the anti-parallel of the angular momentum vector), and \hat{i}_z closes the right-handed system (and represents a “downrange” reference direction). Because of the near impulsivity and the fixed energy of the 2nd stage maneuver (uncontrolled cutoff time), an exact target orbital plane cannot be achieved. Rather, because of the projection term in Equation (2), \hat{i}_z represents a “pseudo” orbital target downrange direction that has been shown to be sufficient for MAV’s orbital plane requirements.

A reorganization of Vis-Viva can be used to compute the desired final velocity magnitude v_d given a current position $r_0 = |\vec{r}_0|$ and desired semi-major axis a_d . Also, r-dot nulling will ensure that the final state will lie entirely along the target orbital downrange basis vector \hat{i}_z . Equipped with this knowledge, the final desired orbital velocity vector \vec{v}_d and the velocity-to-go \vec{v}_{go} can be computed as follows:

$$\vec{v}_d = \sqrt{\mu \left(\frac{2}{r_0} - \frac{1}{a_d} \right)} \hat{i}_z \quad (4)$$

$$\vec{v}_{go} = \vec{v}_d - \vec{v}_0 = \sqrt{\mu \left(\frac{2}{r_0} - \frac{1}{a_d} \right)} \hat{i}_z - \vec{v}_0 \quad (5)$$

Enhanced Inner Loop: Non-Impulsive Maneuvers and Gravity Corrections

The scheme presented above works sufficiently well as a targeting algorithm for MAV in the guided 2nd stage configuration, because MAV can correct for any non-impulsive assumption errors as well as gravity loss errors during the maneuver, which are not accounted for explicitly in Equations (1) and (5). This feedback is not available for an unguided upper stage, so the targeting algorithm must be adjusted to compensate for these errors. Other publications describe methods of compensating for these gravity losses such that the algorithm itself provides a solution that will be more inertially fixed throughout the maneuver¹¹. A method is described here as well, and the results between the two methods have been shown to be near equal in results and computation speed.

Removing the impulsivity assumptions and compensating for gravity losses requires making the inner loop targeting algorithm an iterative cycle, rather than the closed-form Equations (1) through (5). During each inner-loop iteration, this scheme utilizes the estimated velocity-to-go from the previous iteration and produces an updated estimate. The procedure for one iteration is laid out as follows:

- Compute the *estimated* final position and velocity \vec{r}_f and \vec{v}_f after a maneuver is performed, which depends on a *previous* guess of \vec{v}_{go} .
- Given the estimated final position \vec{r}_f , compute the *desired* final velocity \vec{v}_d at that position required to achieve orbital targets.
- Compute the difference between the *estimated* final burn state velocity and the *desired* final burn state velocity. This is the “miss” term $\Delta\vec{v}_{miss}$.
- Use this miss term to update the estimated total velocity-to-go for the next iteration.

The steps above can be written algorithmically as follows, given an initial burn state $[\vec{r}_0, \vec{v}_0]$ and an estimated burn length t_{go} :

1. $\vec{g} = \frac{\mu}{r_0^3} \vec{r}_0$
2. $\vec{v}_f = \vec{v}_0 + \vec{g} t_{go} + \vec{v}_{go_k}$
3. $\vec{r}_f = \vec{r}_0 + \frac{1}{2} \vec{g} t_{go}^2 + \frac{1}{2} (\vec{v}_0 + \vec{v}_f) t_{go}$
4. $\hat{i}_z = \frac{\vec{r}_f \times \hat{i}_y}{|\vec{r}_f \times \hat{i}_y|}$
5. $\vec{v}_d = \sqrt{\mu \left(\frac{2}{r_f} - \frac{1}{a_d} \right)} \hat{i}_z$
6. $\Delta\vec{v}_{miss} = \vec{v}_f - \vec{v}_d$
7. $\vec{v}_{go_{k+1}} = \vec{v}_{go_k} - \Delta\vec{v}_{miss}$

There are a few crucial assumptions that are made in this algorithm modification that allow for the simplifications made. Firstly, in step 1, the gravity vector over the course of the maneuver is just approximated to be the gravity vector based on the initial position vector. Secondly, when computing \vec{r}_f in step 3, a numerical propagation of position can be avoided if the average of the velocity before the maneuver and velocity after the maneuver is used as the velocity term in the position integration. Lastly, this scheme technically assumes a constant acceleration due to thrust, although it has been seen in simulation that a non-constant thrust acceleration profile can be used

with sufficient accuracy for MAV. Using the iteration scheme above, a solution can quickly be found for a \vec{v}_{go} that is inertially constant and compensates for the effect of gravity during the burn.

The Outer Loop: The Predictor

The inner loop described above is responsible for determining the required \vec{v}_{go} at a specific vehicle state to achieve a set of orbital targets. As MAV approaches apoapsis, the solution to this inner-loop scheme at each point along the coast more efficient (costs less Δv). Therefore, the MAV 1st and 2nd stages can be sized together such that, at some point during the ballistic coast phase, the velocity-to-go \vec{v}_{go} will be equal to the energy capability of the vehicle Δv_{cap} . In other words, at some point during the ballistic coast, the following condition is met:

$$|\vec{v}_{go}| = \Delta v_{cap} \quad (6)$$

Once this condition is met, MAV can ignite its second stage and achieve the orbital target desired. The outer-loop algorithm is then responsible for determining when in ballistic coast Equation (6) will be satisfied. Because this equation cannot be solved in closed form, another iterative solution is required.

Initially, a Newton-Raphson (NR) scheme was selected as a possible candidate for the iterator of this algorithm due to the well-behaved profile of $|\vec{v}_{go}|$ as a function of time during MAV's ballistic coast, which can be seen in the "total \vec{v}_{go} " line in the right-hand plot in Figure 2. With the objective being to predict a time in the future at which the MAV should ignite the 2nd stage (referred to as time-to-ignition, or t_{ig}), an NR root function can be derived as:

$$f(t_{ig}) = |\vec{v}_{go}(t_{ig})| - \Delta v_{cap} \quad (7)$$

With the following used as the iteration scheme:

$$t_{ig_{k+1}} = t_{ig_k} - \frac{f(t_{ig_k})}{f'(t_{ig_k})} = t_{ig_k} - \frac{|\vec{v}_{go}(t_{ig_k})| - \Delta v_{cap}}{\frac{\delta}{\delta t} [|\vec{v}_{go}(t_{ig_k})|]} \quad (8)$$

An auxiliary derivation required in Equation (8) is the partial derivative of \vec{v}_{go} with respect to time. A detailed derivation of this can be found in the previous publication on SxS⁴. It is sufficient to prime this iterative scheme with $t_{ig} = 0$ on the first pass.

In a dispersed environment, there may be some cases that underperform during the 1st stage burn so significantly that the required Δv to complete the second maneuver $|\vec{v}_{go}|$ will be greater than Δv_{cap} throughout the entirety of the ballistic coast. In this case, the NR root equation as per Equation (7) does not have a solution, and the iterative scheme in Equation (8) does not converge. Even in cases for which the solution is *near* the root condition, this algorithm becomes inherently unstable. It was decided that a more robust prediction algorithm was required.

Enhanced Outer Loop: Polynomial Prediction Scheme

To combat instability issues near apoapsis, a polynomial scheme was implemented in which a 2nd-order polynomial is constructed that represents $|\vec{v}_{go}|$ as a function of true anomaly f . It is clear from Figure 2 that the behavior of $|\vec{v}_{go}(t_{ig})|$ is ripe for a 2nd-order polynomial representation, and so is $|\vec{v}_{go}(f_{ig})|$. This polynomial is shifted each iteration around the newly predicted burn location until a true anomaly solution is sufficiently stable via some convergence tolerance. After the true anomaly at the burn location is determined, Kepler's two-body equations are used to convert this true anomaly back to t_{ig} .

This updated outer loop scheme addresses several issues that were seen in the NR scheme. First, in underperforming cases where $|\vec{v}_{go}| > \Delta v_{cap}$ throughout the entirety of MAV's ballistic coast, this issue can now be checked in the iterative scheme by operations on the quadratic equation. If this low-energy condition exists, the optimal burn location is set at the minimum point of the 2nd-order polynomial (where the burn is most efficient).

The conversion from iterating on true anomaly rather than time also demonstrated some benefits. In time-space, a secondary “minor” iterative scheme is required to convert from time back to orbital elements at specific locations (alternatively, a propagation scheme could be used rather than Kepler's equations, such as PEG's Super-G algorithm ¹², but the MAV GNC team wanted to find a solution that was more computationally efficient). Basing the outer loop scheme in true-anomaly-space instead of time circumvents this issue, because the transformation from orbital parameters back to time does not require an iterative solution. It is noted that implementing this change in the NR iterative scheme would require taking the partial derivative of $|\vec{v}_{go}|$ with respect to true anomaly rather than time; however, this still does not solve stability issues near apoapsis. The newly formed polynomial scheme can take advantage of this new true anomaly scheme *as well as* stability enhancements near apoapsis.

This enhanced MAV Energy Management algorithm is based on a constantly shifting 2nd-order polynomial which satisfies Equation (6) somewhere along the polynomial. This polynomial takes on the following form:

$$v_{go}(f) = c_0 + c_1 f + c_2 f^2 \quad (9)$$

Throughout this derivation, when v_{go} is seen without vector notation, it is assumed that $v_{go} = |\vec{v}_{go}|$. Equation (9) would satisfy a proper ignition condition when the true anomaly that is input into this function leads to a v_{go} value that matches Δv_{cap} , such as in Equation (6).

To initially prime the algorithm's iterative sequence, three true anomaly values are selected that represent the starting points of the polynomial. The algorithm will adjust these true anomaly values as “bounds” for the polynomial throughout the iteration scheme, so only an approximation of an initial guess is required. The true anomalies for this iteration are labeled f_{0k} , f_{1k} , and f_{2k} . f_{1k} is selected to be the mean of f_{0k} and f_{2k} , so only the outer polynomial bounds f_{0k} and f_{2k} are required as inputs.

To begin the iterative scheme, the orbital elements that represent the current vehicle state are computed, as well as the relevant basis vectors for the orbital perifocal frame ¹³. As will be shown in this algorithm, performing Keplerian position/velocity computations in the perifocal frame avoids the need to perform a lengthier conversion of a full orbital parameter set back to position and velocity vectors.

$$\vec{X} = [\vec{r} \ \vec{v}] \rightarrow a, e, i, \Omega, \omega, f \quad (10)$$

$$h = |\vec{r} \times \vec{v}| \quad (11)$$

$$C_p^I = R_3(-\Omega)R_1(-i)R_3(-\omega) \quad (12)$$

$$\hat{p}^I = C_p^I \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \hat{q}^I = C_p^I \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad (13)$$

Now that these initial parameters and basis vectors are computed, the iterative scheme can begin. Given our three true anomalies that represent our polynomial, values of \vec{v}_{go} are computed at each of these true anomaly locations. This will permit calculation of the relative cost of performing the 2nd stage orbital insertion maneuver at each of these true anomaly locations. To do this, the position and velocity at each true anomaly locations (subscript “i”) can be computed using the perifocal frame basis vectors as follows:

$$\vec{r}_{i_k} = \frac{h^2}{\mu} \frac{1}{1 + e \cos f_{i_k}} (\cos f_{i_k} \hat{p}^I + \sin f_{i_k} \hat{q}^I) \quad , \quad i = 0,1,2 \quad (14)$$

$$\vec{v}_{i_k} = \frac{\mu}{h} [-\sin f_{i_k} \hat{p}^I + (e + \cos f_{i_k}) \hat{q}^I] \quad , \quad i = 0,1,2 \quad (15)$$

$$\vec{v}_{go_{i_k}} = SxS(\vec{r}_{i_k}, \vec{v}_{i_k}, a_d, i_d, \Omega_d, t_{go}, \vec{v}_{go_{i_{k-1}}}) \quad , \quad i = 0, 1, 2 \quad (16)$$

In Equation (16), t_{go} represents a best estimate of the length of the 2nd stage maneuver. As seen in Equation (16), the non-impulsive assumptions and gravity corrections used in the inner loop depend on the previous iteration of \vec{v}_{go} for *that specific index of the polynomial*. This means that each solution to \vec{v}_{go} at the three polynomial locations should be stored in memory so they can be used in the next iteration. For the first iteration, initializing each \vec{v}_{go_k} to the zero vector $\vec{0}$ is sufficient.

Once the final \vec{v}_{go} solutions are computed for the polynomial of the current iteration, the magnitudes $v_{go_{0,1,2}}$ are known (when the subscript k is removed, values are assumed to represent the current iteration). With three true anomaly values $f_{0,1,2}$, and three v_{go} values $v_{go_{0,1,2}}$ computed at those true anomalies, a polynomial can be constructed that will fit through each of these three solution points. This can be done by solving for the following system of equations:

$$v_{go_0} = c_0 + c_1 f_0 + c_2 f_0^2 \quad (17)$$

$$v_{go_1} = c_0 + c_1 f_1 + c_2 f_1^2 \quad (18)$$

$$v_{go_2} = c_0 + c_1 f_2 + c_2 f_2^2 \quad (19)$$

In matrix form, this system of equations can be represented using the Vandermonde matrix ¹⁴ for a 2nd order polynomial as follows:

$$\begin{bmatrix} c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} f_0^2 & f_0 & 1 \\ f_1^2 & f_1 & 1 \\ f_2^2 & f_2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} v_{go_2} \\ v_{go_1} \\ v_{go_0} \end{bmatrix} \quad (20)$$

Solving for the coefficients only requires a simple 3x3 matrix inversion. A closed-form method for solving this system of equations without matrix inversion is used on MAV's flight processor.

Now that the three coefficients are computed, a full 2nd-order polynomial is defined that represents v_{go} magnitude as a function of true anomaly along MAV's ballistic coast. This polynomial inherently compensates for the true non-impulsive nature of the maneuver as well as gravity losses as seen in the inner-loop iteration scheme. Next, the true anomaly at the most "energy efficient" burn location f_m is computed simply as the minimum of the polynomial:

$$f_m = -\frac{c_1}{2c'_0} = -\frac{c_1}{2(c_0 - \Delta v_{cap})} \quad (21)$$

In Equation (22), c'_0 , is used rather than c_0 to ensure that the polynomial solution point that is being solved for passes through Δv_{cap} . Next, the quadratic equation can be used to determine if there is a solution for the desired true anomaly at burn ignition, f_b , other than f_m . If $c_1^2 - 4c_2c'_0 > 0$, then there is a solution that exists somewhere along the coasting trajectory such that $v_{go} = \Delta v_{cap}$. In this case, two 2nd stage burn ignition opportunities exist: one before the minimum energy point, and one after the minimum energy point. MAV's current design is specified to always target the first ignition opportunity in the coast, as follows:

$$f_b = \min \left(\frac{-c_1 + \sqrt{c_1^2 - 4c_2c'_0}}{2c_2} \quad , \quad \frac{-c_1 - \sqrt{c_1^2 - 4c_2c'_0}}{2c_2} \right) \quad (22)$$

If there is no solution outlined above, then f_b is set to f_m . Now that the correct burn location for this iteration was computed, the final v_{go} vector *at this true burn location* can be computed as follows:

$$\vec{r}_b = \frac{h^2}{\mu} \frac{1}{1 + e \cos f_b} (\cos f_b \hat{p}^I + \sin f_b \hat{q}^I) \quad (23)$$

$$\vec{v}_b = \frac{\mu}{h} [-\sin f_b \hat{p}^I + (e + \cos f_b) \hat{q}^I] \quad (24)$$

$$\vec{v}_{go_k} = SxS(\vec{r}_b, \vec{v}_b, a_d, i_d, \Omega_d, t_{go}, \vec{v}_{go_{k-1}}) \quad (25)$$

Notice that once again, the inner loop requires an initial estimate of \vec{v}_{go} . This necessitates that the true solution at the burn location \vec{v}_{go} must also be stored in memory alongside the three solutions for \vec{v}_{go} that exist at the polynomial points. Just as for the polynomial point \vec{v}_{go} solution vectors, initializing the true burn location \vec{v}_{go} to $\vec{0}$ was sufficient.

At this point in the iterative scheme, a new \vec{v}_{go} solution and a future true anomaly are known that satisfy an inertially fixed burn orientation for MAV's 2nd stage. However, it may be seen that the first iteration that $|\vec{v}_{go_k}|$ does not necessarily exactly equal Δv_{cap} . This is because \vec{v}_{go_k} was computed based on a location along a 2nd-order polynomial rather than the true dynamical solution directly at f_b . In other words, although a 2nd order polynomial is a good approximation of how v_{go} changes as a function of true anomaly, the solution only *exactly* matches the polynomial at the three polynomial reference points. Because of this, the iteration scheme must update in such a way that the polynomial becomes localized around the estimated solution location to become more and more accurate each iteration. Therefore, the polynomial bounds f_0, f_1, f_2 are updated for the next iteration around the final solution such that:

$$\begin{bmatrix} f_{0_{k+1}} \\ f_{1_{k+1}} \\ f_{2_{k+1}} \end{bmatrix} = \begin{bmatrix} f_{b_k} - \Delta f \\ f_{b_k} \\ f_{b_k} + \Delta f \end{bmatrix} \quad (26)$$

Where:

$$\Delta f = \max \left(\Delta f_{tol}, \frac{1}{2} (f_{1_k} - f_{0_k}) \right) \quad (27)$$

Once f_b is computed, Keplerian equations can be used to compute the current orbital mean anomaly M , the mean anomaly at the burn location M_b , as well as the mean anomaly at the most optimal location (minimizing Δv cost) M_m . With these values computed, the time-to-ignition parameter t_{ig} , as well as the delta between the time-to-ignition and time-to-minimum-cost Δt_{ig} , can be computed:

$$t_{ig} = \sqrt{\frac{a^3}{\mu}} (M_b - M) \quad (28)$$

$$t_m = \sqrt{\frac{a^3}{\mu}} (M_m - M) \quad (29)$$

$$\Delta t_{ig} = t_m - t_{ig} \quad (30)$$

Now that the values for polynomial true anomalies are updated, the next iteration of $v_{go_{k+1}}$ values can use the current iteration of v_{go_k} values as primers. All parameters are now ready for the next iteration cycle.

The value of Δt_{ig} is a critical parameter for assessing excess energy on-board the 2nd stage. If Δt_{ig} is 0 (i.e. $f_b = f_m$), this means that the Energy Management algorithm predicts that the 2nd stage delta-v available will be insufficient to achieve the intended orbital parameter targets. In this case, guidance ignites 2nd stage near apoapsis to achieve a final state as close to the orbital parameter targets as possible. Conversely, if Δt_{ig} is *too* large, that means the vehicle plans on bleeding off *too much* excess energy into the \dot{r} -nulling effect, which means that the vehicle may be improperly sized for Δv in the 1st and 2nd stage. The desired distribution of Δt_{ig} values would be of all dispersed cases leading to non-zero Δt_{ig} values, but with all Δt_{ig} values *close* to zero. This allows for all dispersed cases to still employ the Energy Management methodology to target the correct orbit, while ensuring that the vehicle is not sized improperly.

Notes on Mechanization and Stability

A full listing of the algorithm code can be found in Appendix A. In previous algorithm versions, a major cycle would run at approximately 1 Hz and iterate the outer loop several times at these 1 Hz intervals until the solution is converged. In this updated scheme, rather than running multiple minor iterations per major 1 Hz cycle, it was shown that running only one iteration per major cycle (essentially eliminating the need to

compute the outer loop multiple times) was shown to have good convergence behavior. This led to a sufficiently accurate solution in just a few cycles running at the 1 Hz major cycle rate, and a significant reduction in total computation time in the GNC subsystem for the flight processor.

Figure 3 shows the energy management solution (left) and solution accuracy (right) evolve over time as the vehicle approaches true burn location for a nominal simulation run of the MAV GNC team’s 6DOF simulation tool. There are several important takeaways from this plot. As seen, starting very early on in the ballistic coast phase, at just 1 Hz iteration cycles, the algorithm was able to compute a stable solution (small velocity error) in a matter of just a few iterations (as seen in the right-hand plot). In this specific run, the algorithm determined that approximately 90 seconds before “minimum energy” condition would be the time at which $\Delta v_{cap} = v_{go}$ (as seen in the left-hand plot). As can be seen in the two plots in Figure 3, although the velocity error (difference between v_{go} and Δv_{cap}) throughout most of the coast phase remains significantly under $1e-5$ m/s of error, Δt_{ig} has a clear and distinct “movement” as the vehicle approaches the most optimal burn location. This is because as the vehicle is exposed to external perturbations (such as atmospheric drag, RCS impulse maneuvers, etc.), the solution slightly shifts. The four visible perturbations in Δt_{ig} (and Δv error) during coast (at times -375s, -225s, -175s, and -75s) are distinct RCS thrusting events that orient the MAV vehicle for fairing maneuver separation (see Figure 1). The longer gradual drift of Δt_{ig} can be characterized as 1) gradual atmospheric drag energy depletion (moving the solution closer to the “minimum energy” burn location (smaller Δt_{ig})), and 2) small inaccuracies of using a two-body Keplerian algorithm under the perturbations of a higher-order gravity model. As seen in the Δv error plot, the RCS events lead to a slight increase in error during the events, but the algorithm is quickly able to correct for this perturbation back down to full algorithm convergence.

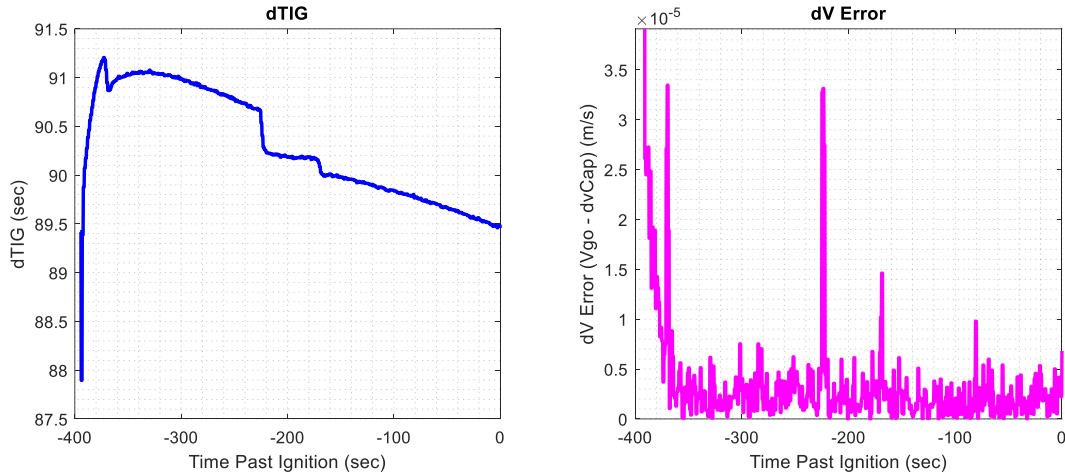


Figure 3: Enhanced Energy Management Algorithm Performance

It was seen during preliminary testing of the algorithm that allowing Δf to shrink infinitely small would lead to precision errors in the polynomial calculation. Therefore, a tolerance is set on the minimum polynomial boundaries of the iterative scheme in Equation (27). This does not limit accuracy of the scheme, however, because the polynomial itself can shift each iteration to localize around the next best estimate of the burn location f_b . In fact, because of the localization shift of the polynomial from Equation (26), it has yet to be seen whether a shrinkage of the polynomial bounds is necessary *at all* because the polynomial solution is “guaranteed” to match the true dynamical solution exactly at the three polynomial boundaries (the middle point at which corresponds to the burn location). In fact, allowing Δf to shrink excessively would mean that the accuracy of the f_m computation may suffer if f_m is sufficiently far away from the polynomial bounds. The MAV GNC team plans on conducting more extensive analysis on the proper bound shifting/shrinking methodology in the future.

In rare cases, it was shown that setting \vec{v}_{go} to $\vec{0}$ for the first iteration still led to an “oscillatory” effect in the solution due to the coupling of the inner loop and outer loop (and specifically, because of the inner loop running only once per outer loop). This led to small pointing adjustments every 1 Hz and eventually a premature RCS depletion. This problem can easily be addressed by running the inner loop some fixed number of times per outer loop computation. It was shown that this extra priming of \vec{v}_{go} helped the solution maintain stability throughout the coast phase and had minimal impact on computational performance. Figure 4 shows the performance of the algorithm for priming the \vec{v}_{go} vector (running the inner loop) multiple times per outer loop. For MAV, it was determined that priming v_{go} only once (running the inner loop only twice each per iteration) led to sufficient accuracy and that future priming would not lead to more accurate results.

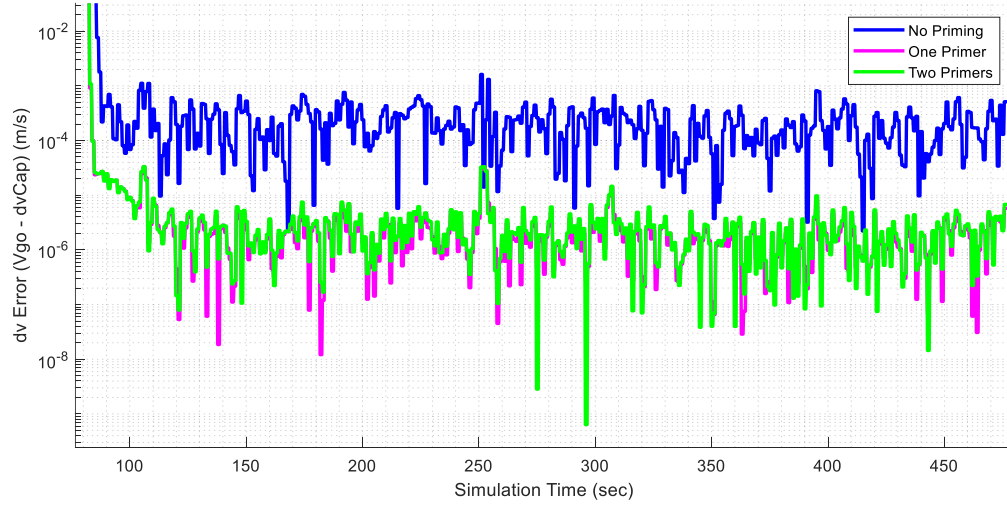


Figure 4: Algorithm Performance for Priming \vec{v}_{go}

In total, the new iteration scheme is robust to the sensitive perturbations of MAV’s ascent profile and does not come with any noticeable hit on computational performance. The MAV team has fully implemented this new algorithm into both the 6DOF simulation test suite and as part of the official deliverable to the MAV FSW team.

ANALYTICAL PARTIALS

Whereas the previous section of this paper discusses algorithm performance for a nominal MAV ascent profile, this section describes a useful tactic that was derived by the MAV GNC team to analyze dispersed ascent profiles for an unguided upper-stage configuration. The MAV vehicle has strict orbital insertion requirements on semi-major axis and final orbital plane to ensure successful rendezvous with the ERO vehicle without significant orbital drift in LMO. During previous design cycles, the MAV GNC team had begun to numerically characterize the sensitivity of different orbital dispersions to different dispersed input parameters. The largest sensitivities to MAV's orbital insertion burn were determined to be mass knowledge error (i.e. Δv_{cap} knowledge error), second stage total impulse error (can also be thought of as Δv_{cap} error), and pointing errors (average pointing errors throughout the 2nd stage burn). These effects were compared to the final semi-major axis and orbital plane (inclination and right-ascension) and were shown to have the greatest effects on orbital insertion dispersion performance.

Because of these sensitivities, the MAV GNC team conducted an analytical study into these dispersions in hopes of finding more physical meaning behind the sensitivities we were seeing. The results of this study, outlined in the sections that follow, provide a set of 1st order analytical partial derivatives of the MAV final orbital insertion parameters with respect to dispersed vehicle inputs, and provide valuable insight as to how expected orbital performance would change given a change in these vehicle inputs. The performance is then compared to the 6DOF simulation suite maintained by the MAV GNC team to verify both the simulation performance as well as the derivation of the partials themselves.

Derivation of the MAV Analytical Partial

Begin by assuming that the vehicle has coasted to the proper insertion location as defined by the Energy Management algorithm, and just before the 2nd stage SRM ignition has the following vehicle state:

$$\vec{X}_0 = \begin{bmatrix} \vec{r}_0 \\ \vec{v}_0 \end{bmatrix} \quad (31)$$

Therefore, nominally, after an assumed impulsive maneuver the nominal vehicle will end at a state:

$$\vec{X}_f = \begin{bmatrix} \vec{r}_0 \\ \vec{v}_f \end{bmatrix} = \begin{bmatrix} \vec{r}_0 \\ \vec{v}_0 + \vec{v}_{go} \end{bmatrix} = \begin{bmatrix} \vec{r}_0 \\ \vec{v}_0 + v_{\eta_0} \hat{v}_{go} \end{bmatrix} \quad (32)$$

where \vec{v}_{go} is calculated using MAV's Energy Management targeting routine. Note that the upper stage will ignite when $|\vec{v}_{go}|$ is equal to the velocity capability of the vehicle. In previous sections, this parameter was identified as Δv_{cap} , but is changed here to v_{η_0} to represent the nominal velocity due to thrust (η), for clarity of the following derivation. Using Tsiolkovsky's rocket equation, the nominal scalar velocity imparted by thrust upon the vehicle is broken down as follows:

$$v_{\eta_0} = g_0 I_{sp_0} \ln \left(\frac{m_0}{m_f} \right) \quad (33)$$

Next, some perturbation parameters are added to this representation that allow for partial derivatives of both impulse uncertainty factor f and mass knowledge error Δm :

$$v_{\eta} = g_0 \left[I_{sp_0} (1 + f) \right] \ln \left(\frac{m_0 + \Delta m}{m_f + \Delta m} \right) \quad (34)$$

The impulse uncertainty term arises because of the actual manufactured impulse of the 2nd stage SRM. The mass knowledge unawareness comes from any mass uncertainties in the 2nd stage vehicle during ascent (due

to dusting, mass measurement errors before launch etc.). There are two more perturbation terms θ_{op} and θ_{ip} that represent the out-of-plane and in-plane average pointing errors during the maneuver, respectively, but they are not found in this scalar representation of v_η in Equation (34) and will be introduced later.

A perturbation frame is used to introduce pointing errors in a more physically meaningful fashion. The perturbation frame's relationship to the Energy Management algorithm's target frame is specified in Figure 5. Note that this is a 2-dimensional representation of the frames, and the frames would be adjusted slightly with any required orbital plane change during the 2nd burn (however, the 2nd stage plane change is small, and this representation is a reasonable approximation for visualization purposes). The basis vectors of this new perturbation frame are defined as follows:

$$\hat{p}_x = \hat{v}_{go} \quad \hat{p}_y = \frac{\vec{v}_0 \times \vec{v}_{go}}{|\vec{v}_0 \times \vec{v}_{go}|} \quad \hat{p}_z = \frac{\hat{p}_x \times \hat{p}_y}{|\hat{p}_x \times \hat{p}_y|} \quad (35)$$

Note that the \hat{p}_x direction aligns with the direction of \vec{v}_{go} , signifying that the vectorized form of Equation (33) can be represented as:

$$\vec{v}_{\eta_0} = v_{\eta_0} \hat{v}_{go} = v_{\eta_0} \hat{p}_x \quad (36)$$

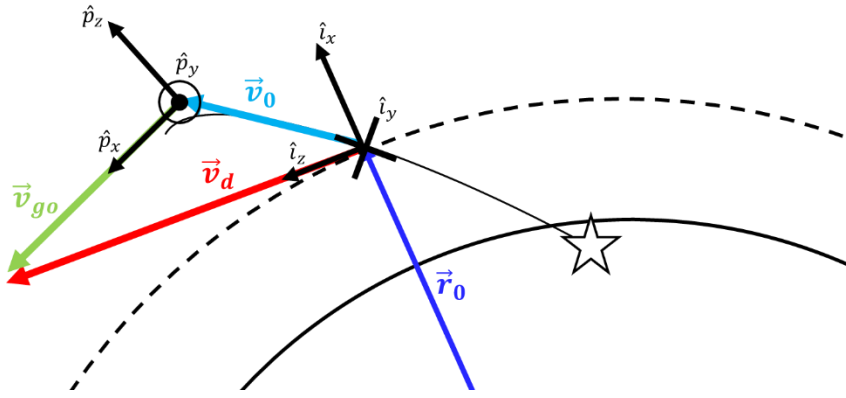


Figure 5: 2-Dimensional Representation of the Perturbation and Target Frames

Using this perturbation frame, we can now construct the components of a complete thrust vector which utilizes the perturbations θ_{op} and θ_{ip} :

$$v_{\eta_x} = \cos \theta_{op} \cos \theta_{ip} v_\eta \quad v_{\eta_y} = v_\eta \sin \theta_{op} \quad v_{\eta_z} = v_\eta \cos \theta_{op} \sin \theta_{ip} \quad (37)$$

Combining all terms, an expanded thrust vector is defined as:

$$\vec{v}_\eta = (1 + f) g_0 I_{sp} \ln \left(\frac{m_0 + \Delta m}{m_f + \Delta m} \right) [(\cos \theta_{op} \cos \theta_{ip}) \hat{p}_x + (\sin \theta_{op}) \hat{p}_y + (\cos \theta_{op} \sin \theta_{ip}) \hat{p}_z] \quad (38)$$

The total thrust vector \vec{v}_η in Equation (38) has more terms than the preliminary version that was provided in Equation (33). However, this form contains properties that allow for partial derivations taken around the four perturbation parameters f , Δm , θ_{op} , and θ_{ip} . The result of taking these partial derivatives is as follows:

$$\frac{\delta}{\delta f} [\vec{v}_\eta] = \frac{v_\eta}{1 + f} [\cos \theta_{op} \cos \theta_{ip} \hat{p}_x + \sin \theta_{op} \hat{p}_y + \cos \theta_{op} \sin \theta_{ip} \hat{p}_z] \quad (39)$$

$$\frac{\delta}{\delta \theta_{op}} [\vec{v}_\eta] = v_\eta [-\sin \theta_{op} \cos \theta_{ip} \hat{p}_x + \cos \theta_{op} \hat{p}_y - \sin \theta_{op} \sin \theta_{ip} \hat{p}_z] \quad (40)$$

$$\frac{\delta}{\delta \theta_{ip}} [\vec{v}_\eta] = v_\eta [-\cos \theta_{op} \sin \theta_{ip} \hat{p}_x + \cos \theta_{op} \cos \theta_{ip} \hat{p}_z] \quad (41)$$

$$\begin{aligned} \frac{\delta}{\delta \Delta m} [\vec{v}_\eta] &= (1+f) g_0 I_{sp} \left[\frac{m_f - m_0}{(m_f + \Delta m)(m_0 + \Delta m)} \right] * \dots \\ &[\cos \theta_{op} \cos \theta_{ip} \hat{p}_x + \sin \theta_{op} \hat{p}_y + \cos \theta_{op} \sin \theta_{ip} \hat{p}_z] \end{aligned} \quad (42)$$

To run a perturbation analysis, these partials are evaluated at their nominal points. For all the defined perturbation parameters, this nominal point was selectively designed to be zero. Evaluating the above partials in Equations (39) through (42) at the nominal zero point for all perturbation parameters:

$$\frac{\delta}{\delta f} [\vec{v}_\eta]_{f, \theta_{op}, \theta_{ip}, \Delta m=0} = v_{\eta_0} \hat{p}_x \quad (43)$$

$$\frac{\delta}{\delta \theta_{op}} [\vec{v}_\eta]_{f, \theta_{op}, \theta_{ip}, \Delta m=0} = v_{\eta_0} \hat{p}_y \quad (44)$$

$$\frac{\delta}{\delta \theta_{ip}} [\vec{v}_\eta]_{f, \theta_{op}, \theta_{ip}, \Delta m=0} = v_{\eta_0} \hat{p}_z \quad (45)$$

$$\frac{\delta}{\delta \Delta m} [\vec{v}_\eta]_{f, \theta_{op}, \theta_{ip}, \Delta m=0} = g_0 I_{sp} \left(\frac{m_f - m_0}{m_0 m_f} \right) \hat{p}_x \quad (46)$$

The benefit of the definition of the perturbation frame is now inherently clear and can be visualized by the simplicity of Equations (43) through (46) as compared to equations (39) through (42). Two perturbation parameters, Δm and f , only affect \vec{v}_η in the \hat{p}_x direction. Then, θ_{op} and θ_{ip} only affect \vec{v}_η in the \hat{p}_y and \hat{p}_z directions, respectively.

Now that partials are derived for the thrust vector with respect to the noted perturbation parameters, it is now beneficial to derive partials for *orbital elements* with respect to our parameters. Now that the partials are derived for \vec{v}_η as seen in Equations (43) through (46), these partials are utilized in the equations to follow. Starting with the Vis-Viva equation:

$$\frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a} \quad (47)$$

A final semi-major axis, a_f , can be represented as a function of final velocity, \vec{v}_f , and position, \vec{r}_0 :

$$a_f = -\frac{\mu}{2 \left(\frac{v_f^2}{2} - \frac{\mu}{r} \right)} = -\frac{1}{2} \mu \left(\frac{v_f^2}{2} - \frac{\mu}{r_0} \right)^{-1} \quad (48)$$

Now, the partial of the final semi-major axis with respect to each of the four perturbation parameters is computed. Note that because of the derivation of the partials of the thrust vector with respect to these perturbation parameters appear in these equations, the same methodology for computing these orbital element partials can be applied *for each perturbation parameter*. Therefore, the partials with respect to some generic perturbation parameter, α , is used, where α can represent any of the perturbation parameters f , Δm , θ_{op} , θ_{ip} .

$$\frac{\delta a_f}{\delta \alpha} = \frac{1}{2} \mu \left(\frac{v_f^2}{2} - \frac{\mu}{r_0} \right)^{-2} v_f \frac{\delta}{\delta \alpha} [|\vec{v}_f|] \quad (49)$$

Using the identity $\frac{\delta |\vec{x}|}{\delta \alpha} = \hat{x} \cdot \frac{\delta \vec{x}}{\delta \alpha}$ (which was also used in the original NR targeting algorithm scheme for \vec{v}_{go} !), coupled with the fact that $\frac{\delta \vec{v}_f}{\delta \alpha} = \frac{\delta \vec{v}_\eta}{\delta \alpha}$, the above can be rewritten into the following form:

$$\frac{\delta a_f}{\delta \alpha} = \frac{1}{2} \mu \left(\frac{v_f^2}{2} - \frac{\mu}{r_0} \right)^{-2} v_f \left[\hat{v}_f \cdot \frac{\delta}{\delta \alpha} [\vec{v}_\eta] \right] \quad (50)$$

Lastly, some simplification can be performed in the first part of Equation (50), the verification of which is left as an exercise to the reader. The simplified partial can be written as:

$$\frac{\delta a_f}{\delta \alpha} = \frac{2a_f^2}{\mu} \left[\vec{v}_f \cdot \frac{\delta}{\delta \alpha} [\vec{v}_\eta] \right] \quad (51)$$

As should be clear by now, the partial derivatives of the orbital insertion parameters with respect to the perturbation parameters becomes quite simple if the equations can be put in terms of \vec{v}_η , for which the partials were already taken. The chain rule is simply used throughout the derivations until the term $\frac{\delta}{\delta \alpha} [\vec{v}_\eta]$ appears and is the only partial derivative that exists in the equation.

This same exercise can be performed for both inclination and RAAN, and the complete derivation for both can be found in Appendix B. For brevity, this derivation is saved for the appendix and the final form of the analytical partials can be found here in Equations (53) through (54):

$$\frac{\delta a_f}{\delta \alpha} = \frac{2a_f^2}{\mu} \left[\vec{v}_f \cdot \frac{\delta}{\delta \alpha} [\vec{v}_\eta] \right] \quad (52)$$

$$\frac{\delta}{\delta \alpha} [i_f] = -\frac{1}{h_f \sin i} \left(\frac{\delta}{\delta \alpha} [\vec{h}_f] - \left(\frac{\delta}{\delta \alpha} [\vec{h}_f] \cdot \hat{h}_f \right) \hat{h}_f \right) \cdot \hat{z} \quad (53)$$

$$\frac{\delta}{\delta \alpha} [\Omega] = -\frac{1}{\sin \Omega} \left(\frac{\delta}{\delta \alpha} [\hat{n}] \cdot \hat{x} \right) \quad (54)$$

Where α can be replaced with f , Δm , θ_{op} , or θ_{ip} . For inclination and RAAN, the auxiliary partial derivatives $\frac{\delta}{\delta \alpha} [\vec{h}_f]$ and $\frac{\delta}{\delta \alpha} [\hat{n}]$ can be found in Appendix B as well.

Finally, these partial derivatives can be used to estimate final orbital dispersion performance of the MAV vehicle by using an additive identity of statistical variances. The individual contributions of each of these perturbation parameters to the total variance of the orbital parameter in question can be simply added together because of the well-known fact that $Var(X + Y) = Var(X) + Var(Y)$ for two independent variables X and Y . These final total variances are then used as an approximation to the total orbital dispersions as shown in Equations (55) through (57):

$$\sigma_a^2 = \left(\frac{\delta a}{\delta f} \sigma_f \right)^2 + \left(\frac{\delta a}{\delta \Delta m} \sigma_{\Delta m} \right)^2 + \left(\frac{\delta a}{\delta \theta_{op}} \sigma_{\theta_{op}} \right)^2 + \left(\frac{\delta a}{\delta \theta_{ip}} \sigma_{\theta_{ip}} \right)^2 \quad (55)$$

$$\sigma_i^2 = \left(\frac{\delta i}{\delta f} \sigma_f \right)^2 + \left(\frac{\delta i}{\delta \Delta m} \sigma_{\Delta m} \right)^2 + \left(\frac{\delta i}{\delta \theta_{op}} \sigma_{\theta_{op}} \right)^2 + \left(\frac{\delta i}{\delta \theta_{ip}} \sigma_{\theta_{ip}} \right)^2 \quad (56)$$

$$\sigma_{\Omega}^2 = \left(\frac{\delta \Omega}{\delta f} \sigma_f \right)^2 + \left(\frac{\delta \Omega}{\delta \Delta m} \sigma_{\Delta m} \right)^2 + \left(\frac{\delta \Omega}{\delta \theta_{op}} \sigma_{\theta_{op}} \right)^2 + \left(\frac{\delta \Omega}{\delta \theta_{ip}} \sigma_{\theta_{ip}} \right)^2 \quad (57)$$

Or, written as a system of equations:

$$\begin{bmatrix} \sigma_a^2 \\ \sigma_i^2 \\ \sigma_{\Omega}^2 \end{bmatrix} = \begin{bmatrix} \left(\frac{\delta a}{\delta f} \right)^2 & \left(\frac{\delta a}{\delta \Delta m} \right)^2 & \left(\frac{\delta a}{\delta \theta_{op}} \right)^2 & \left(\frac{\delta a}{\delta \theta_{ip}} \right)^2 \\ \left(\frac{\delta i}{\delta f} \right)^2 & \left(\frac{\delta i}{\delta \Delta m} \right)^2 & \left(\frac{\delta i}{\delta \theta_{op}} \right)^2 & \left(\frac{\delta i}{\delta \theta_{ip}} \right)^2 \\ \left(\frac{\delta \Omega}{\delta f} \right)^2 & \left(\frac{\delta \Omega}{\delta \Delta m} \right)^2 & \left(\frac{\delta \Omega}{\delta \theta_{op}} \right)^2 & \left(\frac{\delta \Omega}{\delta \theta_{ip}} \right)^2 \end{bmatrix} \begin{bmatrix} \sigma_f^2 \\ \sigma_{\Delta m}^2 \\ \sigma_{\theta_{op}}^2 \\ \sigma_{\theta_{ip}}^2 \end{bmatrix} = \bar{\beta} \bar{s} \quad (58)$$

The power of this new sensitivity matrix, $\bar{\beta}$ should be readily seen by the reader. Given the fact that $\bar{\beta}$ is only dependent on current vehicle sizing parameters and a nominal ascent state, a set of dispersion variances \bar{s} can be multiplied into the vehicle-specific sensitivity matrix $\bar{\beta}$ and an approximation of the orbital parameter dispersions can instantaneously be computed without the need to perform lengthy Monte Carlo simulations. Not only can this be used to verify vehicle performance in simulation but can also be used as a “sizing trade” tool to see how different orbital element dispersions are affected given different vehicle sizes, different ascent profiles, and different dispersion inputs.

Comparison to 6DOF Results

Table 1 shows accuracy results of these 1st order partial derivatives from a recent design cycle of the MAV spin-stabilized upper stage. As seen from the table, the analytical partials slightly underestimate the total expected dispersions of the orbital plane and over-estimate the expected dispersion of orbital semi-major axis. The non-zero errors can be attributed to several factors, mainly higher-order effects that a 1st-order partial derivative ignore but a full Monte Carlo run can accurately capture. However, the surprising accuracy of just the 1st-order derivative shows that just these four input parameters contribute to the majority of MAV’s final orbit dispersions.

Table 1: Analytical Partial Accuracy Compared to a Monte Carlo

| Orbital Parameter | Error (%) |
|-------------------|-----------|
| Semi-Major Axis | 1.912% |
| Inclination | -4.361% |
| RAAN | -3.332% |

CONCLUSION

The introduction of an unguided upper stage posed many problems that the MAV GNC team had to solve to ensure that the MAV was able to still achieve all orbital insertion parameters within requirement bounds. This paper presented an updated version of MAV’s Energy Management targeting algorithm which allows for a spin-stabilized upper stage to target an inertially-fixed orbital insertion maneuver. Notes on mechanization were presented and described as the MAV GNC team began implementing and testing the algorithm. An analytical set of 1st-order partials were also presented that encompass an approximation tool to test sensitivities of dispersed parameters and vehicle sizing on total orbital insertion performance.

The MAV GNC team has also performed several sensitivity and vehicle sizing trades that were not presented directly in this paper in support of the unguided upper-stage redesign. Along with the other studies, the MAV GNC team plans to continue to analyze the effectiveness of both the new targeting algorithm as well as the analytical partials in hopes of ensuring even stronger robustness in future vehicle iterations.

ACKNOWLEDGEMENTS

The authors of this paper would like to acknowledge the entirety of the MAV GNC and 6DOF analysis team for making invaluable contributions to these and other algorithms employed on the MAV vehicle, including but not limited to: Brian Bae, Mike Fritzinger, Joey Hakanson, and Alex Summers. The authors would also like to acknowledge Greg Dukeman, Naeem Ahmad, and Robin Pinson for their expertise in ascent vehicle guidance and CPS algorithms, specifically. Finally, the authors would like to thank Paul Von der Porten, Matthew Hawkins, Tom Fill, and several others that contributed to the SLS-enhanced PEG algorithm from which several ideas were borrowed for MAV's iteration routines.

APPENDIX

Appendix A: Full Listing of the Enhanced Energy Management Algorithm

Below is an entire listing for the enhanced Energy Management algorithm. The entire algorithm below was found to lead to sufficient convergence for MAV if ran at 1 Hz.

- Given: Current Estimated State $[\vec{r}, \vec{v}]$.
- Given: desired orbital target and other constant parameters $a_d, i_d, \Omega_d, t_{go}$.
 - $\hat{l}_y = \begin{bmatrix} -\sin(\Omega_d) \sin(i_d) \\ \cos(\Omega_d) \sin(i_d) \\ -\cos(i_d) \end{bmatrix}$ (target orbital plane)
- If first pass:
 - $\vec{v}_{go_k} = \vec{0}$ (at the burn location)
 - $\vec{v}_{go_{i_k}} = \vec{0}$, $i = 0,1,2$ (at the polynomial point locations)
 - $f_{0_k} = f_0$ (initial guess)
 - $f_{2_k} = f_2$ (initial guess)
 - $f_{1_k} = \frac{1}{2}(f_{0_k} + f_{2_k})$
- Outer Loop:
 - $[\vec{r} \ \vec{v}] \rightarrow a, e, i, \Omega, \omega, f$
 - $h = |\vec{r} \times \vec{v}|$
 - $C_P^I = R_3(-\Omega)R_1(-i)R_3(-\omega)$
 - $\hat{p}^I = C_P^I \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\hat{q}^I = C_P^I \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$
 - For $i = 0,1,2$: (Inner Loop for polynomial points) (**can be computed twice for priming**)
 - $\vec{r}_{i_k} = \frac{h^2}{\mu} \frac{1}{1+e \cos f_{i_k}} (\cos f_{i_k} \hat{p}^I + \sin f_{i_k} \hat{q}^I)$ (position at true anomaly)
 - $\vec{v}_{i_k} = \frac{\mu}{h} [-\sin f_{i_k} \hat{p}^I + (e + \cos f_{i_k}) \hat{q}^I]$ (velocity at true anomaly)
 - $\vec{g} = \frac{\mu}{|\vec{r}_{i_k}|^3} \vec{r}_{i_k}$ (approximation of gravity)
 - $\vec{v}_{f_{i_k}} = \vec{v}_{i_k} + \vec{g} t_{go} + \vec{v}_{go_{i_{k-1}}}$ (estimate final velocity after burn end)
 - $\vec{r}_{f_{i_k}} = \vec{r}_{i_k} + \frac{1}{2} \vec{g} t_{go}^2 + \frac{1}{2} (\vec{v}_{i_k} + \vec{v}_{f_{i_k}}) t_{go}$ (estimate final position at burn end)
 - $\hat{l}_{z_{i_k}} = \frac{\vec{r}_{f_{i_k}} \times \hat{l}_y}{|\vec{r}_{f_{i_k}} \times \hat{l}_y|}$
 - $\vec{v}_{d_{i_k}} = \sqrt{\mu \left(\frac{2}{r_{f_{i_k}}} - \frac{1}{a_d} \right)} \hat{l}_{z_{i_k}}$ (desired velocity at burn end given final position)
 - $\Delta \vec{v}_{miss_i} = \vec{v}_{f_{i_k}} - \vec{v}_{d_{i_k}}$ (missed delta-v)
 - $\vec{v}_{go_{i_k}} = \vec{v}_{go_{i_{k-1}}} - \Delta \vec{v}_{miss_i}$ (update v_{go} estimate at this true anomaly location)
 - Polynomial construction and solution finding:
 - $\begin{bmatrix} c_2 \\ c_1 \\ c_0 \end{bmatrix} = \begin{bmatrix} f_0^2 & f_0 & 1 \\ f_1^2 & f_1 & 1 \\ f_2^2 & f_2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} v_{go_2} \\ v_{go_1} \\ v_{go_0} \end{bmatrix}$
 - $f_m = -\frac{c_1}{2c_0} = -\frac{c_1}{2(c_0 - \Delta v_{cap})}$ (minimum energy true anomaly)
 - If $c_1^2 - 4c_2c_0 > 0$: (check if solutions exist)

- $f_b = \min \left(\frac{-c_1 + \sqrt{c_1^2 - 4c_2c_0'}}{2c_2}, \frac{-c_1 - \sqrt{c_1^2 - 4c_2c_0'}}{2c_2} \right)$
- Otherwise:
 - $f_b = f_m$
- Burn location: (*inner loop for the burn location*) (**can be computed twice for priming**)
 - $\vec{r}_b = \frac{h^2}{\mu} \frac{1}{1+e \cos f_b} (\cos f_b \hat{p}^I + \sin f_b \hat{q}^I)$
 - $\vec{v}_b = \frac{\mu}{h} [-\sin f_b \hat{p}^I + (e + \cos f_b) \hat{q}^I]$
 - $\vec{g} = \frac{\mu}{|\vec{r}_b|^3} \vec{r}_b$
 - $\vec{v}_{f_k} = \vec{v}_b + \vec{g} t_{go} + \vec{v}_{go_{k-1}}$
 - $\vec{r}_{f_k} = \vec{r}_b + \frac{1}{2} \vec{g} t_{go}^2 + \frac{1}{2} (\vec{v}_b + \vec{v}_{f_k}) t_{go}$
 - $\hat{i}_{z_k} = \frac{\vec{r}_{f_k} \times \vec{v}_{f_k}}{|\vec{r}_{f_k} \times \vec{v}_{f_k}|}$
 - $\vec{v}_{d_k} = \sqrt{\mu \left(\frac{2}{r_{f_k}} - \frac{1}{a_d} \right)} \hat{i}_{z_k}$
 - $\Delta \vec{v}_{miss} = \vec{v}_{f_k} - \vec{v}_{d_k}$
 - $\vec{v}_{go_k} = \vec{v}_{go_{k-1}} - \Delta \vec{v}_{miss}$
- Time-to-ignition
 - $f, f_b, f_m \rightarrow M, M_b, M_m$ (*mean anomalies*)
 - $t_{ig} = \sqrt{\frac{a^3}{\mu}} (M_b - M)$ (*time to ignition*)
 - $t_m = \sqrt{\frac{a^3}{\mu}} (M_m - M)$ (*time to minimum energy point*)
 - $\Delta t_{ig} = t_m - t_{ig}$
- Targeting performance parameter (Δv_{err}):
 - $\Delta v_{err} = |\vec{v}_{go_k}| - \Delta v_{cap}$
- Update for next iteration:
 - $\Delta f = \max \left(\Delta f_{tol}, \frac{1}{2} (f_{1_k} - f_{0_k}) \right)$
 - $\begin{bmatrix} f_{0_{k+1}} \\ f_{1_{k+1}} \\ f_{2_{k+1}} \end{bmatrix} = \begin{bmatrix} f_{b_k} - \Delta f \\ f_{b_k} \\ f_{b_k} + \Delta f \end{bmatrix}$ (*re-center polynomial at estimated burn location*)
 - $\vec{v}_{go_{i_{k+1}}} = \vec{v}_{go_{i_k}}, \quad i = 0, 1, 2$ (*update v_{go} values at polynomial points*)
 - $\vec{v}_{go_{k+1}} = \vec{v}_{go_k}$ (*update v_{go} estimate at burn location*)

Appendix B: Analytical Partial Extra Derivation

Starting with the definition of inclination:

$$\vec{h}_f = \vec{r}_0 \times \vec{v}_f \quad (B1)$$

$$i_f = \cos^{-1} \left(\frac{\vec{h}_f}{|\vec{h}_f|} \cdot \hat{z} \right) \quad (B2)$$

And by noting the following:

$$\frac{\delta}{\delta \alpha} [\vec{h}_f] = \vec{r}_0 \times \frac{\delta}{\delta \alpha} [\vec{v}_\eta] \quad (B3)$$

The partial of inclination with respect to our parameters can be defined as follows:

$$\frac{\delta}{\delta \alpha} [i_f] = - \frac{1}{\sqrt{1 - \left(\frac{\vec{h}_f}{|\vec{h}_f|} \cdot \hat{z} \right)^2}} \left[\frac{1}{h_f^2} \left(h_f \frac{\delta}{\delta \alpha} [\vec{h}_f] - \left(\hat{h}_f \cdot \frac{\delta}{\delta \alpha} [\vec{h}_f] \right) \vec{h}_f \right) \right] \cdot \hat{z} \quad (B4)$$

$$\frac{\delta}{\delta \alpha} [i_f] = - \frac{1}{\sqrt{1 - \cos^2 i}} \left[\frac{1}{h_f} \left(\frac{\delta}{\delta \alpha} [\vec{h}_f] - \left(\hat{h}_f \cdot \frac{\delta}{\delta \alpha} [\vec{h}_f] \right) \hat{h}_f \right) \right] \cdot \hat{z} \quad (B5)$$

Leaving finally in a simplified form:

$$\frac{\delta}{\delta \alpha} [i_f] = - \frac{1}{h_f \sin i} \left(\frac{\delta}{\delta \alpha} [\vec{h}_f] - \left(\frac{\delta}{\delta \alpha} [\vec{h}_f] \cdot \hat{h}_f \right) \hat{h}_f \right) \cdot \hat{z} \quad (B6)$$

Next, starting with the following definition of RAAN:

$$\hat{n} = \frac{\hat{z} \times \vec{h}_f}{|\hat{z} \times \vec{h}_f|} \quad (B7)$$

$$\Omega = \cos^{-1}(\hat{n} \cdot \hat{x}) \quad (B8)$$

The partial of RAAN with respect to our parameters can be calculated as follows:

$$\frac{\delta}{\delta \alpha} [\hat{n}] = \frac{1}{|\hat{z} \times \vec{h}_f|^2} \left\{ |\hat{z} \times \vec{h}_f| \left(\hat{z} \times \frac{\delta}{\delta \alpha} [\vec{h}_f] \right) - (\hat{z} \times \vec{h}_f) \left[\hat{n} \cdot \left(\hat{z} \times \frac{\delta}{\delta \alpha} [\vec{h}_f] \right) \right] \right\} \quad (B9)$$

$$\frac{\delta}{\delta \alpha} [\hat{n}] = \frac{1}{|\hat{z} \times \vec{h}_f|} \left\{ \left(\hat{z} \times \frac{\delta}{\delta \alpha} [\vec{h}_f] \right) - \left[\hat{n} \cdot \left(\hat{z} \times \frac{\delta}{\delta \alpha} [\vec{h}_f] \right) \right] \hat{n} \right\} \quad (B10)$$

$$\frac{\delta}{\delta \alpha} [\Omega] = - \frac{1}{\sin \Omega} \left(\frac{\delta}{\delta \alpha} [\hat{n}] \cdot \hat{x} \right) \quad (B11)$$

REFERENCES

- ¹ Yaghoubi, Schnell. “Mars Ascent Vehicle Hybrid Propulsion Configuration”, *2020 IEEE Aerospace Conference*, March 2020
- ² Yaghoubi, Schnell. “Mars Ascent Vehicle Solid Propulsion Configuration”, *2020 IEEE Aerospace Conference*, March 2020
- ³ Yaghoubi, Ma. “Integrated Design Results for the MSR DAC-0.0 Mars Ascent Vehicle”, *2021 IEEE Aerospace Conference*, March 2021.
- ⁴ Everett, Jason M. “A Generalized Approach to In-Space Solid-Propellant Vehicle Maneuvers”, AAS 20-118, February 2020
- ⁵ Hoag, David. “Apollo Guidance, Navigation and Control Systems”, *MIT Instrumentation Library*, December 1968
- ⁶ Richard H. Battin. “An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition”, *American Institute of Aeronautics & Astronautics*, 1999
- ⁷ Anzalone, Erickson, Everett, Powers. “Guidance and Navigation Challenges for a Mars Ascent Vehicle”, *2020 IEEE Aerospace Conference*, March 2020
- ⁸ Xu, Chen. “An Energy Management Ascent Guidance Algorithm for Solid Rocket-Powered Launch Vehicles”, *17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*, April 2011
- ⁹ Si-Yuan Chen, Qun-Li Xia. “A Multiconstrained Ascent Guidance Method for Solid Rocket-Powered Launch Vehicles”, *International Journal of Aerospace Engineering*, Volume 2016
- ¹⁰ R. L. McHenry. “Space Shuttle Ascent Guidance, Navigation, and Control”, *The Journal of the Astronautical Sciences*, Vol. XXVII, No. 1, pp. 1-38, 1979
- ¹¹ Brand, Timothy. “A New Approach to Lambert Guidance”, June 1971
- ¹² Pinson, Von der Porten, Ahmad. “Space Launch System Guidance Description for Artemis I”, NASA Marshall Space Flight Center, 2019
- ¹³ Bate, Mueller, White. “Fundamentals of Astrodynamics”
- ¹⁴ Turner, L. Richard. “Inverse of the Vandermonde Matrix with Applications”, *NASA Technical Note*, TN D-3547